Self-Supervised Learning in Vision

Lecture 6

Rob Fergus

Slides from Ishan Misra, Naiyan Wang & many others.
Success story of supervision

ImageNet Challenge

Classification Results (CLS)

- 2010: 0.28
- 2011: 0.26
- 2012: 0.16
- 2013: 0.12
- 2014: 0.07
- 2015: 0.036
- 2016: 0.03
- 2017: 0.023

- 16.7% ↓
- 23.3% ↓
Success story of supervision: Pre-training

- Features from networks pre-trained on ImageNet can be used for a variety of different downstream tasks.
Success story of supervision: Recipe for good solutions

• Pre-train on a large supervised dataset.
• Collect a dataset of "supervised" images
• Train a ConvNet
Can we get labels for all data?
Can we get labels for all data?

Stats from Pawan Kumar at Oxford
Can we get labels for all data?
Can we get labels for all data?

Can we get labels for all data?

Bounding Boxes  | Image Level  | Internet Photos

Internet Photos have significantly more data compared to Bounding Boxes and Image Level.
Can we get labels for all data?

ImageNet (14 million images) needed 22 human years to label
Can we get labels for all data?

- What about complex concepts?
  - Video?

- Labelling cannot scale to the size of the data we generate
Rare concepts?

10% of the classes account for 93% of the data

Slide credit: Rob Fergus
Different Domains?

ImageNet pre-training may not work
Arguments for Unsupervised Learning

• Want to be able to exploit unlabeled data
  • Vast amount of it often available
  • Essentially free

• Good regularizer for supervised learning
  • Helps generalization
  • Transfer learning
  • Zero/one/few - shot learning
Unsupervised Learning

• Biological argument [from G. Hinton]:
  • Our brains have $10^{15}$ connections
  • We live for $10^9$ secs
  • Need $10^6$ bits/sec
  • Insufficient information from occasional high level label
  • Only source with enough information is input itself

• Challenging problem: big focus on many DL groups
**Historical Note**

- Deep Learning revival started in ~2006
  - Hinton & Salakhudinov Science paper on RBMs

- **Unsupervised** Deep Learning was focus from 2006-2012

- In ~2012 great results in vision, speech with **supervised** methods appeared
  - Initially less interest in unsupervised learning
  - By focus once more on unsupervised learning
Overview of Unsupervised Perspectives

• Given just data \{X\}
  • Unlike supervised learning there are no provided labels \{Y\}

1. Density modeling, i.e. build model of p(X)
   • Enables sampling of new data
   • Evaluate probability of a data point
   • Can be conditional model, e.g. p(X_t \mid X_{\{t-1\},...})
   • Requires (deep) generative architectures [HARD]
   • Generating pixels not necessarily optimal for learning representations for downstream tasks
1. Density Modeling

- Have access to $x \sim p_{data}(x)$ through training set

- Want to learn a model $x \sim p_{model}(x)$

- Want $p_{model}$ to be similar to $p_{data}$:

Samples from true data distribution have high likelihood under $p_{model}$

Samples drawn from $p_{model}$ reflect structure of $p_{data}$
2. “Self supervised” learning

• Also unsupervised but...

• Find supervision signal $y$ **within the input data**

• This signal is then used as a target in *discriminative model*:

$$y : \mathcal{X} \rightarrow \mathcal{Y}$$

$$x \mapsto y(x)$$

• Allows the use of standard supervised learning losses and architectures

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(f_{\theta}(x_i), y(x_i))$$

• Pre-training of representation for subsequent task

• Typically involves some insight into domain to pick $y$
Class Project

- Everyone must come and discuss proposed ideas with me at some point during office hours.
- Simple option is to reimplement existing approach/paper.
  - Cannot just copy existing repo, but can use it for debugging.
- Alternatively, you can extend existing approach/paper.
  - Can use existing repo, but extension must be significant
- More ambitious (more marks): Implement new idea from scratch
  - Can use existing models/code but must have significant novelty
- Project abstracts due Oct 15th (email to me).
  - Email paragraph summarizing your project plan to fergus@cs.nyu.edu.
  - Please be sure to include the names of people on your project.
Class Project details

Team of 2-3 people [1 person not allowed; 3 is hard max]
PyTorch preferred

Project video:
- 2 minute clip explaining your project
- Voice over slides

Project Report
- Format: 4-8 page conference paper style report on your project (please don't waffle)
- Intro (with refs to related work) [~1 page]
- Method (be sure to cite any code/pre-trained models) [~2 pages]
- Experiments (must have plots/results figures; also should have baselines; ideally some kind of ablation experiments too) [~2-4 pages]
- Discuss (brief) [~0.5 pages]
- See examples: http://openaccess.thecvf.com/CVPR2018.py
- Zip of source code or link to Github (please ensure you give access to robfergus)
Class Project details (2)

- Deadline is Thursday Dec 17th midnight [Hard deadline]
  - Feel free to turn in earlier
  - Will try to grade them and compute final grades by Christmas

- Grading (49% of total grade for class)
  - Novelty / Technical difficulty of problem [15%]
  - Quality of Results [15%]
  - Quality of implementation [5%]
  - Quality of writeup & video presentation [14%]
  - How many people in your group
Class Project General Advice

- Please make sure you have *something* working, even if you don’t achieve overall goal
- Even a small part of an ambitious project can be OK
- So **please have a safe plan B option** in mind
- Expect all projects to train something, i.e. must use b-prop at some point
- Just evaluating existing models is NOT OK.
- Cluster gets busy at end of semester -- please don’t leave it all to last moment.
Self-supervised learning in computer vision

Ishan Misra

With slides from Andrew Zisserman, Carl Doersch
What is "self" supervision?

- Obtain labels from the data itself by using a "semi-automatic" process
What is "self" supervision?

- Obtain labels from the data itself by using a "semi-automatic" process.
Word2vec

- Fill in the blanks
Success of self-supervised learning in NLP

- Fill in the blanks is a powerful signal to learn representations

- Sentence/Word representations: BERT - Devlin et al., 2018
Why self supervision?

- Helps us learn using observations and interactions
- Does not require exhaustive annotation of concepts
- Leverage multiple modalities or structure in the domain
Range of self-supervised systems

- Doersch et al. (2015)
- Zhang et al. (2016)
- Zhang et al. (2017)
- Noroozi et al. (2016)
- Pathak et al. (2016)
Range of self-supervised systems

- Doersch et al. (2015)
- Zhang et al. (2016)
- Zhang et al. (2017)
- Noroozi et al. (2016)
- Pathak et al. (2016)

- Wang et al. (2015)
- Misra et al. (2016)
- Pathak et al. (2017)
Range of self-supervised systems

- Doersch et al. (2015)
- Zhang et al. (2016)
- Zhang et al. (2017)
- Noroozi et al. (2016)
- Pathak et al. (2016)

- Wang et al. (2015)
- Misra et al. (2016)
- Pathak et al. (2017)

- Owens et al. (2016)
- Zhang et al. (2017)
- Bansal et al. (2016)
Range of self-supervised systems

- Images
  - Doersch et al. (2015)
  - Zhang et al. (2016)
  - Zhang et al. (2017)
  - Norouzi et al. (2016)
  - Pathak et al. (2016)

- Videos
  - Wang et al. (2015)
  - Misra et al. (2016)
  - Pathak et al. (2017)
  - Owens et al. (2016)
  - Zhang et al. (2017)

- Sound & depth
  - Bansal et al. (2016)
  - Agarwal et al. (2015)
  - Jayaraman et al. (2015)
  - Pinto et al. (2016)
  - Agarwal et al. (2016)
  - Pinto et al. (2016)
  - Pinto et al. (2017)
Self-supervision in computer vision

• Using images
• Using video
• Using video and sound
From Images: Relative position of patches

8 possible locations

Randomly Sample Patch
Sample Second Patch

Unsupervised visual representation learning by context prediction, Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015
From Images: Relative position of patches

Unsupervised visual representation learning by context prediction, Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015
Relative Position: Nearest Neighbors in features

<table>
<thead>
<tr>
<th>Input</th>
<th>Relative-positioning</th>
<th>Random Initialization</th>
<th>ImageNet AlexNet</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Input Images" /></td>
<td><img src="image2" alt="Relative-positioning Images" /></td>
<td><img src="image3" alt="Random Initialization Images" /></td>
<td><img src="image4" alt="ImageNet AlexNet Images" /></td>
</tr>
</tbody>
</table>

What do we learn when we solve a Jigsaw puzzle?
Noorozi et al. (2016)

Hash Set

<table>
<thead>
<tr>
<th>index</th>
<th>table</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>9,4,6,8,3,2,5,1,7</td>
</tr>
</tbody>
</table>

Reorder patches according to the selected hash table

[Noorozi et al. (2016)]
Solving a puzzle means predicting the index of the hash table.

[Noorozi et al. (2016)]
• Visualization of filters
Feature Learning by Inpainting

[Context Encoders: Feature Learning by Inpainting, Pathak et al. (2016)]

[Pathak et al. (2016)]
Context Encoders

- Encoder can be substituted with any network architecture like AlexNet etc.
- Decoder is a set of UpConv/deconv/frac-strided-conv layers

[Pathak et al. (2016)]
Combined L2 + GAN loss

Input Image  L2 Loss  Adversarial Loss  Joint Loss

[Pathak et al. (2016)]
Image colorization

Colorful Image Colorization
Richard Zhang, Phillip Isola, Alexei (Alyosha) Efros

http://richzhang.github.io/colorization/
Grayscale image: $L$ channel

$X \in \mathbb{R}^{H \times W \times 1}$

Color information: $ab$ channels

$\hat{Y} \in \mathbb{R}^{H \times W \times 2}$
Grayscale image: $X \in \mathbb{R}^{H \times W}$

Semantics? Higher-level abstraction?

$\mathcal{F}$

Concatenate $(L, ab)$

"Free" supervisory signal
From Images: Predicting Rotations

Which image has the correct rotation?
[Dosovitskiy et al. ICLR 2014]

- 1 class = single image + its transformations
- Learn to classify each “class”
- Domain knowledge about appropriate transformations
- does not scale
Many different self-supervision tasks, how to evaluate?
Self-supervised pre-training

Pre-train data → ConvNet → Learn a representation → Position/Colorization
Fine-tune on end task (Image Classification)

Tests representation as well as how good the pre-training initialization is
Fine-tune on end task

<table>
<thead>
<tr>
<th>Initialization (ResNet101)</th>
<th>End task</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ImageNet top-5 accuracy</td>
</tr>
<tr>
<td>ImageNet Supervised</td>
<td>85.1</td>
</tr>
<tr>
<td>Relative Position</td>
<td>59.2</td>
</tr>
<tr>
<td>Colorization</td>
<td>62.5</td>
</tr>
</tbody>
</table>

- Multi-task self-supervised visual learning, C Doersch, A Zisserman, ICCV 2017
Are they complementary?

<table>
<thead>
<tr>
<th>Initialization (ResNet101)</th>
<th>End task</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ImageNet top-5 accuracy</td>
</tr>
<tr>
<td>ImageNet Supervised</td>
<td>85.1</td>
</tr>
<tr>
<td>Relative Position</td>
<td>59.2</td>
</tr>
<tr>
<td>Colorization</td>
<td>62.5</td>
</tr>
<tr>
<td>Relative Position + Colorization (Multi-task)</td>
<td>66.6</td>
</tr>
</tbody>
</table>

- Multi-task self-supervised visual learning, C Doersch, A Zisserman, ICCV 2017
Train linear classifiers on "fixed" features

Tests how good the representation is (linearly separable)
Self-supervision in computer vision

- Using images
- Using video
- Using video and sound
Video

- Video is a "sequence" of frames
- How to get "self-supervision"?
- Predict order of frames
- Fill in the blanks
- Track objects and predict their position
Video

- Slow feature
  - Neighborhood frames should have similar features

\[
\mathcal{U}_2 = \{((j, k), p_{jk}) : \mathbf{x}_j, \mathbf{x}_k \in \mathcal{U} \text{ and } p_{jk} = 1(0 \leq j - k \leq T)\},
\]

\[
\begin{align*}
R_2(\theta, \mathcal{U}) &= \sum_{(j, k) \in \mathcal{U}_2} D_\delta(z_{\theta}(\mathbf{x}_j), z_{\theta}(\mathbf{x}_k), p_{jk}) \\
&= \sum_{(j, k) \in \mathcal{U}_2} p_{jk} d(z_{\theta_j}, z_{\theta_k}) + \overline{p_{jk}} \max(\delta - d(z_{\theta_j}, z_{\theta_k}), 0),
\end{align*}
\]

Video

- Slow and steady feature
  - Not only similar, but also smooth
  - Extend to triplet setting (Not triplet loss!)

\[ \mathcal{U}_3 = \left\{ \langle (l, m, n), p_{lmn} \rangle : x_l, x_m, x_n \in \mathcal{U} \text{ and } p_{lmn} = 1(0 \leq m - l = n - m \leq T) \right\}. \]

\[ R_3(\theta, \mathcal{U}) = \sum_{(l, m, n) \in \mathcal{U}_3} D_\delta(z_{\theta_l} - z_{\theta_m}, z_{\theta_m} - z_{\theta_n}, p_{lmn}). \]

From video: shuffle and learn

Temporally Correct order

Original video

Temporally Incorrect order
From video: shuffle and learn

Given a start and an end, can this point lie in between?
From video: shuffle and learn

Input Tuple

Correct/Incorrect Tuple

Cross Entropy Loss

fc8

concatenation

classification

From video: shuffle and learn

Nearest Neighbors of Query Frame (fc7 features)

Query

ImageNet

Shuffle & Learn

Random

Central Jamaica
From video: shuffle and learn

Fine-tune on Human Keypoint Estimation
## From video: shuffle and learn

### Fine-tune on Human Keypoint Estimation

<table>
<thead>
<tr>
<th>Initialization (AlexNet)</th>
<th>End task</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FLIC Dataset Keypoints</strong></td>
<td><strong>MPII Dataset Keypoints</strong></td>
</tr>
<tr>
<td>ImageNet Supervised</td>
<td>AUC</td>
</tr>
<tr>
<td></td>
<td>51.3</td>
</tr>
<tr>
<td>Shuffle and Learn</td>
<td>49.6</td>
</tr>
</tbody>
</table>

From video: encoding more structure

Odd-one-out networks - Fernando, Bilen, Gavves, Gould in ICCV 2017

Idea: Object Tracking in Videos

[Wang & Gupta 2015]
Approach

• Use object tracking in videos

• Classify if patches belong to the same track or not

[Wang & Gupta 2015]
Patch Mining In Videos

• Track 8M patches in 100K videos from YouTube.

• Use off-the-shelf tracking algorithms with no learning.

[Wang & Gupta 2015]
VOC 2007 Detection Performance
(pretraining for R-CNN)

% Average Precision

- ImageNet Layout: 68.6
- Tracking: 61.7
- No Pretraining: 60.5
- VGG (16-layer): 42.4

[Wang & Gupta 2015]
Object Movement

• The world is rigid, or at least piecewise rigid
  • Motion provide evidence of how pixels move together
  • The pixels move together are likely to form an object

Self-supervision in computer vision

- Using images
- Using video
- Using video and sound
Audio-Visual co-supervision

Train a network to predict if image and audio clip correspond

Correspond?
Audio-Visual co-supervision

“Objects that Sound”, Arandjelović and Zisserman, ICCV 2017 & ECCV 2018
Audio-Visual co-supervision

“Objects that Sound”, Arandjelović and Zisserman, ICCV 2017 & ECCV 2018
Audio-Visual co-supervision

What can be learnt?

• Good representations – Visual features
  – Audio features
  • Intra- and cross-modal retrieval
  – Aligned audio and visual embeddings
  • “What is making the sound?”
  – Learn to localize objects that sound

“Objects that Sound”, Arandjelović and Zisserman, ICCV 2017 & ECCV 2018
Audio-Visual co-supervision

What would make this sound?

Note, no video (motion) information is used

“Objects that Sound”, Arandjelović and Zisserman, ICCV 2017 & ECCV 2018
Visual + Audio

[Ambient Sound Provides Supervision for Visual Learning, [Owens et al. (2016)]
Unit visualizations
Top responses (unit #90 of 256)
Audio label

Unit visualizations

256 filters

Conv 5

376 conv 5 256 13
Cross-Modality

• Ego-motion
  • “We move in order to see and we see in order to move” - J.J. Gibson
  • Ego-motion data is easy to collect
  • Siamese CNN to predict camera translation & Rotation along 3-axises. (Visual)

Agrawal, P., Carreira, J., & Malik, J. Learning to see by moving. In ICCV 2015

[A Survey to Self-Supervised Learning, Naiyan Wang]
Cross-Modality

• Ego-motion
  • Learning features that are equivariant to ego-motion

Jayaraman, D., & Grauman, K. Learning image representations tied to ego-motion. In *ICCV 2015*

[A Survey to Self-Supervised Learning, Naiyan Wang]
Cross-Modality

• Ego-motion
  • Siamese networks with contrastive loss
  • $M_g$ is the transformation matrix specified by the external sensors

\[
(\theta^*, M^*) = \arg \min_{\theta, M} \sum_{g, i, j} d_g(M_g z_\theta(x_i), z_\theta(x_j), p_{ij}),
\]

\[
d_g(a, b, c) = \mathbb{1}(c = g)d(a, b) + \mathbb{1}(c \neq g)\max(\delta - d(a, b), 0),
\]

Jayaraman, D., & Grauman, K. Learning image representations tied to ego-motion. In ICCV 2015

[A Survey to Self-Supervised Learning, Naiyan Wang]
Cross-Modality

• Acoustics -> RGB
  • Similar events should have similar sound.
  • Naturally cluster the videos.

Cross-Modality

• Features for grasping
  • Verify whether we could grasp the center of a patch at a given angle

Pinto, L., & Gupta, A. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In ICRA 2016
Evaluation

- Evaluate on general high-level vision tasks (classification, detection)
  - Be caution of different settings!

<table>
<thead>
<tr>
<th>Method</th>
<th>Full train set</th>
<th>150 image set</th>
<th>#wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All</td>
<td>&gt;c1</td>
<td>&gt;c2</td>
</tr>
<tr>
<td>Supervised</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Imagenet</td>
<td>56.5</td>
<td>57.0</td>
<td>57.1</td>
</tr>
<tr>
<td>Sup. Masks (Ours)</td>
<td>51.7</td>
<td>51.8</td>
<td>52.7</td>
</tr>
<tr>
<td>Unsupervised</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jigsaw↑[30]</td>
<td>49.0</td>
<td>50.0</td>
<td>48.9</td>
</tr>
<tr>
<td>Kmeans[23]</td>
<td>42.8</td>
<td>42.2</td>
<td>40.3</td>
</tr>
<tr>
<td>Egomotion[2]</td>
<td>37.4</td>
<td>36.9</td>
<td>34.4</td>
</tr>
<tr>
<td>Inpainting[35]</td>
<td>39.1</td>
<td>36.4</td>
<td>34.1</td>
</tr>
<tr>
<td>Tracking-gray[46]</td>
<td>43.5</td>
<td>44.6</td>
<td>44.6</td>
</tr>
<tr>
<td>Sounds[33]</td>
<td>42.9</td>
<td>42.3</td>
<td>40.6</td>
</tr>
<tr>
<td>BiGAN[10]</td>
<td>44.9</td>
<td>44.6</td>
<td>44.7</td>
</tr>
<tr>
<td>Colorization[51]</td>
<td>44.5</td>
<td>44.9</td>
<td>44.7</td>
</tr>
<tr>
<td>Split-Brain Auto[52]</td>
<td>43.8</td>
<td>45.6</td>
<td>45.6</td>
</tr>
<tr>
<td>Context[8]</td>
<td>49.9</td>
<td>48.8</td>
<td>44.4</td>
</tr>
<tr>
<td>Context-videos[8]</td>
<td>47.8</td>
<td>47.9</td>
<td>46.6</td>
</tr>
<tr>
<td>Motion Masks (Ours)</td>
<td>48.6</td>
<td>48.2</td>
<td>48.3</td>
</tr>
</tbody>
</table>


[A Survey to Self-Supervised Learning, Naiyan Wang]
Main issue with all these methods

• All these models rely on expert knowledge

• Need to define $y(x)$ for each new domain

• Not clear how to select a $y(x)$ that is a good target to learn all-purpose features
Unsupervised Learning by Predicting Noise

Piotr Bojanowski, Armand Joulin

ICML 2017
Unsupervised Learning by Predicting Noise

[Bojanowski & Joulin, ICML 2017]

- Inspired by Dosovitskiy et al.
- Learn mapping from images to a sphere
- Fix targets on sphere
- Simultaneously:
  - Learn the mapping
  - Optimize the assignment between images and targets
Deep Discriminative Clustering

• We are given a set of $n$ images

\[ \{x_1, \ldots, x_n\} \]

• We want to learn a visual features $f$ without using labels

\[
\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \min_{y_i} \ell(f_{\theta}(x_i), y_i)
\]

• We use the L2 loss

\[
\min_{\theta} \min_{Y} \frac{1}{2n} \| f_{\theta}(X) - Y \|_F^2
\]

[Bojanowski & Joulin, ICML 2017]
Label Collapse Problem

• Optimization over Y would lead to a collapse

• Repulsive costs are tricky to use

• Can impose constraints on Y but hard to optimize

[Bojanowski & Joulin, ICML 2017]
Fixing the Target Representation

• Instead, we fix the target representation
• Allow a reassignment between targets and images

\[ Y = PC \]

\[ \mathcal{P} = \{ P \in \{0, 1\}^{n \times k} \mid P1 = 1, P^T 1 = 1 \} \]

• Targets C are uniformly sampled on the sphere

\[ \min_\theta \min_{P \in \mathcal{P}} \frac{1}{2n} \| f_\theta(X) - PC \|_F^2 \]

• Final objective function

[Bojanowski & Joulin, ICML 2017]
Optimization

• We minimize our cost function in an on-line fashion
• We use the following algorithm:

\[
\begin{align*}
\textbf{Require: } & T \text{ batches of images, } \lambda_0 > 0 \\
& \text{for } t = \{1, \ldots, T\} \text{ do} \\
& \quad \text{Obtain batch } b \text{ and representations } r \\
& \quad \text{Compute } f_\theta(X_b) \\
& \quad \text{Compute } P^* \text{ by minimizing w.r.t. } P \\
& \quad \text{Compute } \nabla_\theta L(\theta) \text{ using } P^* \\
& \quad \text{Update } \theta \leftarrow \theta - \lambda_t \nabla_\theta L(\theta) \\
\text{end for}
\end{align*}
\]

[Bojanowski & Joulin, ICML 2017]
Optimizing the Permutation Matrix

• At theta fixed, the permutation is obtained by solving

\[
\max_{P \in \mathcal{P}} \text{Tr} \left( P C f_\theta(X)^\top \right).
\]

• Which is a linear program on the set of permutation matrices

• We can use the Hungarian algorithm

\[\mathcal{O}(nb^2)\]

[Bojanowski & Joulin, ICML 2017]
Experimental Setup

• AlexNet architecture

• Learn unsupervised features on ImageNet training set

• Retrain a classifier on top for a target transfer task, i.e. PASCAL VOC Classification / Detection

[Bojanowski & Joulin, ICML 2017]
Baselines

• Self supervised models
  • Wang & Gupta – Temporal coherence in videos
  • Doersch et al. – Predict context patches
  • Zhang et al. – Predict color
  • Norouzi & Favaro – Solve jigsaw puzzles

• Unsupervised model
  • GAN
  • Auto-encoder
  • BI-GAN (Donahue et al.)

[Bojanowski & Joulin, ICML 2017]
Pascal VOC - results

<table>
<thead>
<tr>
<th></th>
<th>Classification</th>
<th>Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trained layers</td>
<td>fc6-8</td>
<td>all</td>
</tr>
<tr>
<td>ImageNet labels</td>
<td>78.9</td>
<td>79.9</td>
</tr>
<tr>
<td>Agrawal et al.</td>
<td>31.0</td>
<td>54.2</td>
</tr>
<tr>
<td>Pathak et al.</td>
<td>34.6</td>
<td>56.5</td>
</tr>
<tr>
<td>Wang &amp; Gupta</td>
<td>55.6</td>
<td>63.1</td>
</tr>
<tr>
<td>Doersch et al.</td>
<td>55.1</td>
<td>65.3</td>
</tr>
<tr>
<td>Zhang et al.</td>
<td>61.5</td>
<td>65.6</td>
</tr>
<tr>
<td>Autoencoder</td>
<td>16.0</td>
<td>53.8</td>
</tr>
<tr>
<td>GAN</td>
<td>40.5</td>
<td>56.4</td>
</tr>
<tr>
<td>BiGAN</td>
<td>52.3</td>
<td>60.1</td>
</tr>
<tr>
<td>NAT</td>
<td>56.7</td>
<td>65.3</td>
</tr>
</tbody>
</table>

- compare favorably to SOTA
- Poor performance of AE / GAN

[Bojanowski & Joulin, ICML 2017]
Nearest Neighbor Queries

[Bojanowski & Joulin, ICML 2017]
Bojanowski & Joulin Summary

• Simple unsupervised approach

• No domain expert knowledge

• Scales to very large datasets

• Close to supervised pipeline

• SOTA performance (at the time) amongst unsupervised methods
Contrastive Predictive Coding

Aaron van den Oord, Yazhe Li, Oriol Vinyals
Google DeepMind

Model Overview

Although this figure shows audio as input, we use the same setup for images, text and reinforcement learning. We hypothesize that these approaches are fruitful partly because the context from which we predict related values are often conditionally dependent on the same shared high-level latent information. And by casting this as a prediction problem, we automatically infer these features of interest to representation learning.

In this paper we propose the following: first, we compress high-dimensional data into a much more compact latent embedding space in which conditional predictions are easier to model. Secondly, we use powerful autoregressive models in this latent space to make predictions many steps in the future. Finally, we rely on Noise-Contrastive Estimation for the loss function in similar ways that have been used for learning word embeddings in natural language models, allowing for the whole model to be trained end-to-end. We apply the resulting model, Contrastive Predictive Coding (CPC) to widely different data modalities, images, speech, natural language and reinforcement learning, and show that the same mechanism learns interesting high-level information on each of these domains, outperforming other approaches.

2 Contrastive Predictive Coding

We start this section by motivating and giving intuitions behind our approach. Next, we introduce the architecture of Contrastive Predictive Coding (CPC). After that we explain the loss function that is based on Noise-Contrastive Estimation. Lastly, we discuss related work to CPC.

2.1 Motivation and Intuitions

The main intuition behind our model is to learn the representations that encode the underlying shared information between different parts of the (high-dimensional) signal. At the same time it discards low-level information and noise that is more local. In time series and high-dimensional modeling, approaches that use next step prediction exploit the local smoothness of the signal. When predicting further in the future, the amount of shared information becomes much lower, and the model needs to infer more global structure. These 'slow features' that span many time steps are often more interesting (e.g., phonemes and intonation in speech, objects in images, or the story line in books).

One of the challenges of predicting high-dimensional data is that unimodal losses such as mean-squared error and cross-entropy are not very useful, and powerful conditional generative models which need to reconstruct every detail in the data are usually required. But these models are computationally intense, and waste capacity at modeling the complex relationships in the data. For example, images may contain thousands of bits of information while the high-level latent variables such as the class label contain much less information (10 bits for 1,024 categories). This suggests that modeling \( p(x|c) \) directly may not be optimal for the purpose of extracting shared information between \( x \) and \( c \). When predicting future information we instead encode the target \( x_{(future)} \) and context \( c_{(present)} \) into a compact distributed vector representations (via non-linear...
CPC Principle

• Encode the target \( x \) (future) and context \( c \) (present) into a compact distributed vector representations (via non-linear learned mappings) in a way that maximally preserves the mutual information of the original signals \( x \) and \( c \) defined as

\[
I(x; c) = \sum_{x,c} p(x, c) \log \frac{p(x|c)}{p(x)}.
\]

• By maximizing the mutual information between the encoded representations (which is bounded by the MI between the input signals), we extract the underlying latent variables the inputs have in common.

By using a density ratio estimation, one can pool the representations from either randomly sampled negative values. Estimation or Importance Sampling could help improve results further.

For simplicity we opted for standard architectures such as strided convolutional layers with resnet blocks for the encoder, and GRUs for the decoder. Finally, note that any type of encoder and autoregressive model can be used in the proposed framework.

As argued in the previous section we do not predict future observations but instead maximize the mutual information between the encoded representations (which is bounded by the MI between the input signals), we extract the underlying latent variables the inputs have in common.

By using an architecture that maximally preserves the mutual information of the original signals, we might instead estimate the density ratio in equation 2. This can be done by using a simple log-bilinear model:

\[
\text{loss} = \sum_{t} \log \frac{f(x_{t}, c_{t})}{\sum_{k \neq t} f(x_{t}, c_{k})}.
\]

Alternatively, non-linear networks or recurrent neural networks could be used. Optimizing this loss will result in learned mappings in a way that maximally preserves the mutual information of the original signals x and c defined as

\[
x_{t} = f_{t}(x_{t-1}, c_{t}),
\]

where \( f \) is the latent space and produces a context latent representation potentially with a lower temporal resolution. Next, an autoregressive model maps the input sequence of observations to a sequence of latent representations (via non-linear functions). In the proposed model, either of the prediction methods are used for the prediction with a different context latent representation.

To capture phonetic content. In other cases, where no additional context is required, one can pool the representations from either randomly sampled negative values. Estimation or Importance Sampling could help improve results further.
CPC Model

- Do NOT predict future observations $x_{t+k}$ directly with a generative model $p_k(x_{t+k} | c_t)$
- Instead we model a **density ratio** which preserves the mutual information between $x_{t+k}$ and $c_t$ (prev eqn):

$$f_k(x_{t+k}, c_t) \propto \frac{p(x_{t+k} | c_t)}{p(x_{t+k})}$$

Where: $$f_k(x_{t+k}, c_t) = \exp \left( z_{t+k}^T W_k c_t \right),$$
Noise Contrastive Estimation Loss

• Given a set $X = \{x_1, \ldots, x_N\}$ of $N$ random samples containing one positive sample from $p(x_{t+k} | c_t)$ and $N - 1$ negative samples from the ‘proposal’ distribution $p(x_{t+k})$, we optimize

$$
\mathcal{L}_N = - \mathbb{E}_X \left[ \log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]
$$

Optimizing this loss will result in $f_k(x_{t+k}, c_t)$ estimating the density ratio in prev slide.

• Not always clear in expts where the $N$ random samples come from
  • E.g. same/different sequence? Narrow window?
C.F. Vision SSL approaches

Unsupervised Visual Representation Learning by Context Prediction
[Doersch et al. ICCV 2015]
CPC applied to Audio

Table 1: LibriSpeech phone and speaker classification results. For phone classification there are 41 possible classes and for speaker classification 251. All models used the same architecture and the same audio input sizes.

<table>
<thead>
<tr>
<th>Method</th>
<th>ACC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Phone classification</strong></td>
<td></td>
</tr>
<tr>
<td>Random initialization</td>
<td>27.6</td>
</tr>
<tr>
<td>MFCC features</td>
<td>39.7</td>
</tr>
<tr>
<td>CPC</td>
<td>64.6</td>
</tr>
<tr>
<td>Supervised</td>
<td>74.6</td>
</tr>
<tr>
<td><strong>Speaker classification</strong></td>
<td></td>
</tr>
<tr>
<td>Random initialization</td>
<td>1.87</td>
</tr>
<tr>
<td>MFCC features</td>
<td>17.6</td>
</tr>
<tr>
<td>CPC</td>
<td>97.4</td>
</tr>
<tr>
<td>Supervised</td>
<td>98.5</td>
</tr>
</tbody>
</table>

Table 2: LibriSpeech phone classification ablation experiments. More details can be found in Section 3.1.

<table>
<thead>
<tr>
<th>Method</th>
<th>ACC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>#steps predicted</strong></td>
<td></td>
</tr>
<tr>
<td>2 steps</td>
<td>28.5</td>
</tr>
<tr>
<td>4 steps</td>
<td>57.6</td>
</tr>
<tr>
<td>8 steps</td>
<td>63.6</td>
</tr>
<tr>
<td>12 steps</td>
<td>64.6</td>
</tr>
<tr>
<td>16 steps</td>
<td>63.8</td>
</tr>
<tr>
<td><strong>Negative samples from</strong></td>
<td></td>
</tr>
<tr>
<td>Mixed speaker</td>
<td>64.6</td>
</tr>
<tr>
<td>Same speaker</td>
<td>65.5</td>
</tr>
<tr>
<td>Mixed speaker (excl.)</td>
<td>57.3</td>
</tr>
<tr>
<td>Same speaker (excl.)</td>
<td>64.6</td>
</tr>
<tr>
<td>Current sequence only</td>
<td>65.2</td>
</tr>
</tbody>
</table>

Figure 2: t-SNE visualization of audio (speech) representations for a subset of 10 speakers (out of 251). Every color represents a different speaker.

Figure 3: Average accuracy of predicting the positive sample in the contrastive loss for 1 to 20 latent steps in the future of a speech waveform. The model predicts up to 200ms in the future as every step consists of 10ms of audio.
CPC applied to images

Additionally, Figure 2 shows a t-SNE visualization for the model predictions showing the importance of predicting multiple steps. With a linear classifier trained on these features, the model shows how linearly separable the data is.

The ImageNet dataset has been used to evaluate unsupervised vision models by many authors. Table 2 gives an overview of two ablation studies of CPC for phone classification. In the first set of experiments, we explore the impact of the window size (maximum context size) for the GRU. A longer window size has a big impact on the performance, and longer segments would give better results.

For speaker voice-characteristics, it is important to note that the window size (maximum context size) has a direct effect on the performance. Interestingly, CPCs capture both speaker identity and speech contents, as demonstrated by the good accuracies attained with a simple linear classifier, which also gets close to the oracle, fully unsupervised approach.

Table 5: Classification accuracy on five common NLP benchmarks. We follow the same transfer learning setup from Skip-thought vectors and use a ResNet v2 101 architecture as the image encoder. After unsupervised training, a linear layer is trained to measure how discriminative the embeddings are and use a ResNet v2 101 architecture as the image encoder. The fully supervised model represents from a random initialized model (i.e., a single learned layer). The results are shown in Table 1 (top). We compare the accuracy with three baselines:

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1 ACC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using AlexNet conv5</td>
<td></td>
</tr>
<tr>
<td>Video [27]</td>
<td>29.8</td>
</tr>
<tr>
<td>Relative Position</td>
<td>30.4</td>
</tr>
<tr>
<td>[11]</td>
<td></td>
</tr>
<tr>
<td>BiGan [34]</td>
<td>34.8</td>
</tr>
<tr>
<td>Colorization [10]</td>
<td>35.2</td>
</tr>
<tr>
<td>Jigsaw [28] *</td>
<td>38.1</td>
</tr>
<tr>
<td>Using ResNet-V2</td>
<td></td>
</tr>
<tr>
<td>Motion Segmentation</td>
<td>27.6</td>
</tr>
<tr>
<td>[35]</td>
<td></td>
</tr>
<tr>
<td>Exemplar [35]</td>
<td>31.5</td>
</tr>
<tr>
<td>Relative Position</td>
<td>36.2</td>
</tr>
<tr>
<td>[35]</td>
<td></td>
</tr>
<tr>
<td>Colorization [35]</td>
<td>39.6</td>
</tr>
<tr>
<td>CPC</td>
<td><strong>48.7</strong></td>
</tr>
</tbody>
</table>

Table 3: ImageNet top-1 unsupervised classification results. *Jigsaw is not directly comparable to the other AlexNet results because of architectural differences.
SimCLR: A Simple Framework for Contrastive Learning of Visual Representations

Ting Chen
Simon Kornblith
Mohammad Norouzi
Geoffrey Hinton

Google Research, Brain Team
The proposed SimCLR framework

A simple idea: maximizing the agreement of representations under data transformation, using a contrastive loss in the latent/feature space.

Figure 2. A framework for contrastive representation learning. Two separate stochastic data augmentations $t, t' \sim \mathcal{T}$ are applied to each example to obtain two correlated views. A base encoder network $f(\cdot)$ with a projection head $g(\cdot)$ is trained to maximize agreement in latent representations via a contrastive loss.
The proposed SimCLR framework

We use random crop and color distortion for augmentation.

Examples of augmentation applied to the left most images:
The proposed SimCLR framework

$f(x)$ is the base network that computes internal representation.

We use (unconstrained) ResNet in this work. However, it can be other networks.
The proposed SimCLR framework

\( g(h) \) is a projection network that project representation to a latent space.

We use a 2-layer non-linear MLP (fully connected net).
The proposed SimCLR framework

Maximize agreement using a contrastive task:

Given \( \{x_k\} \) where two different examples \( x_i \) and \( x_j \) are a positive pair, identify \( x_j \) in \( \{x_k\}_{k \neq i} \) for \( x_i \).

Loss function:

Let \( \text{sim}(u, v) = \frac{u^\top v}{\|u\| \|v\|} \)

\[
\ell_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j) / \tau)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp(\text{sim}(z_i, z_k) / \tau)}
\]
Data Augmentation for Contrastive Representation Learning
We study a set of transformations...

Systematically study a set of augmentation

(a) Original  
(b) Crop and resize  
(c) Crop, resize (and flip)  
(d) Color distort. (drop)  
(e) Color distort. (jitter)

(f) Rotate \{90^\circ, 180^\circ, 270^\circ\}  
(g) Cutout  
(h) Gaussian noise  
(i) Gaussian blur  
(j) Sobel filtering

* Note that we only test these for ablation, the augmentation policy used to train our models only involves random crop (with flip and resize) + color distortion + Gaussian blur.
Composition of augmentations are crucial
Composition of crop and color stands out!

Figure 5. Linear evaluation (ImageNet top-1 accuracy) under individual or composition of data augmentations, applied only to one branch. For all columns but the last, diagonal entries correspond to single transformation, and off-diagonals correspond to composition of two transformations (applied sequentially). The last column reflects the average over the row.

Figure 6. Histograms of pixel intensities (over all channels) for different crops of two different images (i.e. two rows). The image for the first row is from Figure 4. All axes have the same range.
Encoder and Projection Head
A nonlinear projection head improves the representation quality of the layer before it.

We compare three projection heads $g(.)$ (after average pooling of ResNet):

- Identity mapping
- Linear projection
- Nonlinear projection with one additional hidden layer (and ReLU activation)

*Figure 8. Linear evaluation of pretraining with different projection heads. The dimension of $h$ (before projection) is 2048.*
A nonlinear projection head improves the representation quality of the layer before it

To understand why this happens, we measure information in $h$ and $z=g(h)$

<table>
<thead>
<tr>
<th>What to predict?</th>
<th>Random guess</th>
<th>Representation $h$</th>
<th>Representation $g(h)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color vs grayscale</td>
<td>80</td>
<td>99.3</td>
<td>97.4</td>
</tr>
<tr>
<td><strong>Rotation</strong></td>
<td>25</td>
<td>67.6</td>
<td>25.6</td>
</tr>
<tr>
<td>Orig. vs corrupted</td>
<td>50</td>
<td>99.5</td>
<td>59.6</td>
</tr>
<tr>
<td>Orig. vs Sobel filtered</td>
<td>50</td>
<td>96.6</td>
<td>56.3</td>
</tr>
</tbody>
</table>

*Table 3.* Accuracy of training additional MLPs on different representations to predict the transformation applied. Other than crop and color augmentation, we additionally and independently add rotation (one of \{0°, 90°, 180°, 270°\}), Gaussian noise, and Sobel filtering transformation during the pretraining for the last three rows. Both $h$ and $g(h)$ are of the same dimensionality, i.e. 2048.
Loss Function and Batch Size
Normalized cross entropy loss with adjustable temperature works.

<table>
<thead>
<tr>
<th>Name</th>
<th>Negative loss function</th>
<th>Gradient w.r.t. $u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NT-Xent</td>
<td>$u^T v^+ / \tau - \log \sum_{v \in {v^+, v^-}} \exp(u^T v / \tau)$</td>
<td>$(1 - \frac{\exp(u^T v^+ / \tau)}{Z(u)}) / \tau v^+ - \sum_v - \frac{\exp(u^T v^- / \tau)}{Z(u)} / \tau v^-$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Margin</th>
<th>NT-Logi.</th>
<th>Margin (sh)</th>
<th>NT-Logi.(sh)</th>
<th>NT-Xent</th>
</tr>
</thead>
<tbody>
<tr>
<td>50.9</td>
<td>51.6</td>
<td>57.5</td>
<td>57.9</td>
<td>63.9</td>
</tr>
</tbody>
</table>

*Table 4. Linear evaluation (top 1) for models trained with different loss functions. “sh” means using semi-hard negative mining.*
NT-Xent loss needs N and T

<table>
<thead>
<tr>
<th>Name</th>
<th>Negative loss function</th>
<th>Gradient w.r.t. ( u )</th>
</tr>
</thead>
<tbody>
<tr>
<td>NT-Xent</td>
<td>( u^T v^+ / \tau - \log \sum_{v \in {v^+, v^-}} \exp (u^T v / \tau) ) ( (1 - \frac{\exp(u^T v^+/\tau)}{Z(u)}) / \tau v^+ - \sum_{v^-} \frac{\exp(u^T v^-/\tau)}{Z(u) / \tau} v^- )</td>
<td>( u^T v^+ / \tau - \log \sum_{v \in {v^+, v^-}} \exp (u^T v / \tau) ) ( (1 - \frac{\exp(u^T v^+/\tau)}{Z(u)}) / \tau v^+ - \sum_{v^-} \frac{\exp(u^T v^-/\tau)}{Z(u) / \tau} v^- )</td>
</tr>
</tbody>
</table>

- L2 normalization with temperature scaling makes a better loss.
- Contrastive accuracy is not correlated with linear evaluation when l2 norm and/or temperature are changed.

<table>
<thead>
<tr>
<th>( \ell_2 ) norm?</th>
<th>( \tau )</th>
<th>Entropy</th>
<th>Contrast. task acc.</th>
<th>Top 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>0.05</td>
<td>1.0</td>
<td>90.5</td>
<td>59.7</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>4.5</td>
<td>87.8</td>
<td>64.4</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>8.2</td>
<td>68.2</td>
<td>60.7</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>8.3</td>
<td>59.1</td>
<td>58.0</td>
</tr>
<tr>
<td>No</td>
<td>10</td>
<td>0.5</td>
<td>91.7</td>
<td>57.2</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.5</td>
<td>92.1</td>
<td>57.0</td>
</tr>
</tbody>
</table>

*Table 5.* Linear evaluation for models trained with different choices of \( \ell_2 \) norm and temperature \( \tau \) for NT-Xent loss. The contrastive distribution is over 4096 examples.
Contrastive learning benefits from larger batch sizes and

*Figure 9.* Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.
Linear evaluation
7% relative improvement over previous SOTA (cpc v2), matching fully-supervised ResNet-50.

<table>
<thead>
<tr>
<th>Method</th>
<th>Architecture</th>
<th>Param.</th>
<th>Top 1</th>
<th>Top 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Agg.</td>
<td>ResNet-50</td>
<td>24</td>
<td>60.2</td>
<td>-</td>
</tr>
<tr>
<td>MoCo</td>
<td>ResNet-50</td>
<td>24</td>
<td>60.6</td>
<td>-</td>
</tr>
<tr>
<td>PIRL</td>
<td>ResNet-50</td>
<td>24</td>
<td>63.6</td>
<td>-</td>
</tr>
<tr>
<td>CPC v2</td>
<td>ResNet-50</td>
<td>24</td>
<td>63.8</td>
<td>85.3</td>
</tr>
<tr>
<td>Ours</td>
<td>ResNet-50</td>
<td>24</td>
<td>69.3</td>
<td>89.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods using other architectures:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotation</td>
</tr>
<tr>
<td>BigBiGAN</td>
</tr>
<tr>
<td>AMDIM</td>
</tr>
<tr>
<td>CMC</td>
</tr>
<tr>
<td>MoCo</td>
</tr>
<tr>
<td>CPC v2</td>
</tr>
<tr>
<td>Ours</td>
</tr>
<tr>
<td>Ours</td>
</tr>
</tbody>
</table>

Table 6. ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.

Figure 1. ImageNet top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet). Our method, SimCLR, is shown in bold.
Semi-supervised learning

10% relative improvement over previous SOTA (cpc v2), outperforms AlexNet with 100X fewer labels.

<table>
<thead>
<tr>
<th>Method</th>
<th>Architecture</th>
<th>Label fraction 1%</th>
<th>Label fraction 10%</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Methods using other label-propagation:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pseudo-label</td>
<td>ResNet50</td>
<td>51.6</td>
<td>82.4</td>
</tr>
<tr>
<td>VAT+Entropy Min.</td>
<td>ResNet50</td>
<td>47.0</td>
<td>83.4</td>
</tr>
<tr>
<td>UDA (w. RandAug)</td>
<td>ResNet50</td>
<td>-</td>
<td>88.5</td>
</tr>
<tr>
<td>FixMatch (w. RandAug)</td>
<td>ResNet50</td>
<td>-</td>
<td>89.1</td>
</tr>
<tr>
<td>S4L (Rot+VAT+En. M.)</td>
<td>ResNet50 (4×)</td>
<td>-</td>
<td>91.2</td>
</tr>
<tr>
<td><strong>Methods using representation learning only:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>InstDisc</td>
<td>ResNet50</td>
<td>39.2</td>
<td>77.4</td>
</tr>
<tr>
<td>BigBiGAN</td>
<td>RevNet-50 (4×)</td>
<td>55.2</td>
<td>78.8</td>
</tr>
<tr>
<td>PIRL</td>
<td>ResNet-50</td>
<td>57.2</td>
<td>83.8</td>
</tr>
<tr>
<td>CPC v2</td>
<td>ResNet-161(*)</td>
<td>77.9</td>
<td>91.2</td>
</tr>
<tr>
<td>Ours</td>
<td>ResNet-50</td>
<td>75.5</td>
<td>87.8</td>
</tr>
<tr>
<td>Ours</td>
<td>ResNet-50 (2×)</td>
<td>83.0</td>
<td>91.2</td>
</tr>
<tr>
<td>Ours</td>
<td>ResNet-50 (4×)</td>
<td><strong>85.8</strong></td>
<td><strong>92.6</strong></td>
</tr>
</tbody>
</table>

*Table 7. ImageNet accuracy of models trained with few labels.*
Transfer learning

When fine-tuned, SimCLR significantly outperforms the supervised baseline on 5 datasets, whereas the supervised baseline is superior on only 2*. On the remaining 5 datasets, the models are statistically tied.

<table>
<thead>
<tr>
<th></th>
<th>Food</th>
<th>CIFAR10</th>
<th>CIFAR100</th>
<th>Birdsnap</th>
<th>SUN397</th>
<th>Cars</th>
<th>Aircraft</th>
<th>VOC2007</th>
<th>DTD</th>
<th>Pets</th>
<th>Caltech-101</th>
<th>Flowers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linear evaluation:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-supervised</td>
<td>76.9</td>
<td>95.3</td>
<td>80.2</td>
<td>48.4</td>
<td>65.9</td>
<td>60.0</td>
<td>61.2</td>
<td>84.2</td>
<td>78.9</td>
<td>89.2</td>
<td>93.9</td>
<td>95.0</td>
</tr>
<tr>
<td>Supervised</td>
<td>75.2</td>
<td>95.7</td>
<td>81.2</td>
<td>56.4</td>
<td>64.9</td>
<td>68.8</td>
<td>63.8</td>
<td>83.8</td>
<td>78.7</td>
<td>92.3</td>
<td>94.1</td>
<td>94.2</td>
</tr>
<tr>
<td><strong>Fine-tuned:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-supervised</td>
<td>89.4</td>
<td>98.6</td>
<td>89.0</td>
<td>78.2</td>
<td>68.1</td>
<td>92.1</td>
<td>87.0</td>
<td>86.6</td>
<td>77.8</td>
<td>92.1</td>
<td>94.1</td>
<td>97.6</td>
</tr>
<tr>
<td>Supervised</td>
<td>88.7</td>
<td>98.3</td>
<td>88.7</td>
<td>77.8</td>
<td>67.0</td>
<td>91.4</td>
<td>88.0</td>
<td>86.5</td>
<td>78.8</td>
<td>93.2</td>
<td>94.2</td>
<td>98.0</td>
</tr>
<tr>
<td>Random init</td>
<td>88.3</td>
<td>96.0</td>
<td>81.9</td>
<td>77.0</td>
<td>53.7</td>
<td>91.3</td>
<td>84.8</td>
<td>69.4</td>
<td>64.1</td>
<td>82.7</td>
<td>72.5</td>
<td>92.5</td>
</tr>
</tbody>
</table>

*The two datasets, where the supervised ImageNet pretrained model is better, are Pets and Flowers, which share a portion of labels with ImageNet.

Table 8. Comparison of transfer learning performance of our self-supervised approach with supervised baselines across 12 natural image classification datasets, for ResNet-50 (4×) models pretrained on ImageNet. Results not significantly worse than the best (p > 0.05, permutation test) are shown in bold. See Appendix B.6 for experimental details and results with standard ResNet-50.
Conclusions

SimCLR is a simple yet effective self-supervised learning framework, advancing state-of-the-art by a large margin.

- The superior performance of SimCLR is not due to any single design choice, but a combination of design choices.
- Our studies reveal several important factors that enable effective representation learning, which could help future research.

Code & checkpoints available in [github.com/google-research/simclr](https://github.com/google-research/simclr).
References


References

- **Colorization:**

- **Optical Flow**

- **Others**
  - Pinto, L., Gandhi, D., Han, Y., Park, Y. L., & Gupta, A. The curious robot: Learning visual representations via physical interactions. In *ECCVW 2016*. 
What happens when we "super-size" self-supervised learning?

Scaling and benchmarking self-supervised visual representation learning -

Bits of information

• For ImageNet we have ~1M images and 1000 classes
  • Each image has $\log_2(1000)$ bits of information
  • Total = $1M \times \log_2(1000)$

• For self-supervised methods we have ~1M images
  • Each image has lesser bits of information - $\log_2(B)$
  • Total = $1M \times \log_2(B)$
  • **Increase 1M to 100M?**
  • **Increase B to 10B?**
How large scale?

- Focus on existing **popular image-based** self-supervised methods
- Scale along three axes
Specific self-supervision problems

Jigsaw puzzles
(Noorozi & Favaro, 2016)

Colorization
(Zhang & Efros, 2016)

Images from the ImageNet dataset
Jigsaw Puzzles

- Use $N=9$ patches
- In practice, use a subset of permutations
- E.g. 100 from 9!
- Each patch is processed independently
- N-way ConvNet (shared params)

- Problem Complexity
  - Size of subset
Colorization

- Predict N=313 colors
  - Binning LAB space
  - Multi-modal loss

- Problem Complexity
  - Number of colors
  - Number of neighbors in multi-modal loss

At each pixel, classify which color bins
How large scale?

- Scale two techniques - Jigsaw and Colorization
- Scale along three axes

Problem Complexity "Difficulty"

Models
AlexNet, ResNet-50

Data Size
YFCC - 100M
Evaluating the representation

Extract "fixed" features

ConvNet
"Investigation" task

- Train a Linear SVM on **fixed feature** representations
"Investigation" task

- Train a Linear SVM on **fixed feature** representations
- Use the VOC07 image classification task
Scaling on Data Axis

Problem Complexity
"Difficulty"

Models
AlexNet, ResNet-50

Data Size
YFCC - 100M
Scaling on Data Axis

Gain for ResNet50: 10 points

Gain for AlexNet: 2 points
Scaling on Data Axis

Gain for ResNet50: **12** points

Gain for AlexNet: **8** points
Scaling on Problem Complexity

Problem Complexity
"Difficulty"

Models
AlexNet, ResNet-50

Data Size
YFCC - 100M
Scaling on Problem Complexity Axis

Gain for ResNet50: 6 points
Gain for AlexNet: 2 points
Scaling on Problem Complexity Axis

Colorization – VOC07 Linear SVM

mAP

Number K in soft-encoding

ResNet50

AlexNet
Scaling on Data and Problem Complexity

- "Difficulty"

- Problem Complexity
- Models: AlexNet, ResNet-50
- Data Size: YFCC - 100M
Gains along both data and problem axes are complementary.
Our Evaluation – many tasks

- Image classification
  - Few-shot learning
  - ImageNet, Places-205, VOC’07, COCO

- Object detection
  - VOC’07

- 3D Understanding
  - Surface Normals – NYUv2

- Navigation
  - Gibson environment

Images from the Places, VOC07, NYUv2 and Gibson datasets
Our Evaluation – fine-tuning vs. linear classifier

Fine-tune all layers

Linear classifier

A good representation transfers with *little training*
Object Detection

Image classification
Few-shot learning

Object detection
VOC’07

3D Understanding
Surface Normals – NYUv2

Navigation
Gibson environment
Object Detection

- **Fast R-CNN** (Girshick et al., 2015)
  - Same optimization parameters for all methods (including supervised)
  - No “bells and whistles”
  - Use VOC’07
## Object Detection

**VOC07 test set.**

**Fast R-CNN ResNet50**

<table>
<thead>
<tr>
<th>Initialization</th>
<th>Train Set</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VOC07</td>
<td>VOC07+12</td>
<td></td>
</tr>
<tr>
<td>ImageNet Supervised</td>
<td>70.5</td>
<td>76.2</td>
<td></td>
</tr>
<tr>
<td>Places Supervised</td>
<td>67.2</td>
<td>74.5</td>
<td></td>
</tr>
<tr>
<td>Jigsaw ImageNet-1k</td>
<td>61.4</td>
<td>68.3</td>
<td></td>
</tr>
<tr>
<td>Jigsaw ImageNet-22k</td>
<td>69.2</td>
<td>75.4</td>
<td></td>
</tr>
<tr>
<td>Jigsaw YFCC100M</td>
<td>66.6</td>
<td>73.3</td>
<td></td>
</tr>
</tbody>
</table>

*Fine-tune all within error*
Object Detection - Training RoI heads only

VOC07 test set.

Fast R-CNN ResNet50

<table>
<thead>
<tr>
<th>Initialization</th>
<th>Train Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VOC07</td>
</tr>
<tr>
<td>ImageNet Supervised</td>
<td>68.5</td>
</tr>
<tr>
<td>Places Supervised</td>
<td>65.3</td>
</tr>
<tr>
<td>Jigsaw ImageNet-1k</td>
<td>56.6</td>
</tr>
<tr>
<td>Jigsaw ImageNet-22k</td>
<td>67.1</td>
</tr>
<tr>
<td>Jigsaw YFCC100M</td>
<td>62.3</td>
</tr>
</tbody>
</table>

within error

Last layer
Surface Normal Estimation

Image classification
Few-shot learning
ImageNet, Places-205, VOC’07, COCO

Object detection
VOC’07

3D Understanding
Surface Normals – NYUv2

Navigation
Gibson environment
Surface Normal Estimation

• Predict surface normals on NYU-v2
  • Same optimization parameters for all methods (including supervised)
  • PSPNet Architecture
  • Train last few layers only (res5 onwards)

Input

Output

Image from the NYU dataset
<table>
<thead>
<tr>
<th>Initialization</th>
<th>Median Error</th>
<th>% correct within 11.25°</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageNet Supervised</td>
<td>17.1</td>
<td>36.1</td>
</tr>
<tr>
<td>Places Supervised</td>
<td>14.2</td>
<td>41.8</td>
</tr>
<tr>
<td>Jigsaw ImageNet-1k</td>
<td>14.5</td>
<td>41.2</td>
</tr>
<tr>
<td>Jigsaw ImageNet-22k</td>
<td>13.4</td>
<td>43.7</td>
</tr>
<tr>
<td>Jigsaw YFCC100M</td>
<td>13.1</td>
<td>44.6</td>
</tr>
</tbody>
</table>
Surface Normal Estimation

Image classification
Few-shot learning
ImageNet, Places-205, VOC’07, COCO

Object detection
VOC’07

3D Understanding
Surface Normals – NYUv2

Navigation
Gibson environment
Visual Navigation

- Visual navigation in the Gibson environment
  - Method from Sax et al., 2018
  - Fixed ConvNet features
Visual Navigation
Few-shot learning
Places-205, VOC’07

Object detection
VOC’07

3D Understanding
Surface Normals – NYUv2

Navigation
Gibson environment
Few shot learning

- k-shot learning
  - Use VOC’07/Places205 classification
  - K labeled examples per class
  - Train linear SVMs

Image from the Places dataset
Few shot learning

Self-supervised representations are not as sample efficient
Few shot learning - VOC07

VOC07 low-shot conv1

VOC07 low-shot conv2

VOC07 low-shot conv3

VOC07 low-shot conv4

VOC07 low-shot conv5
Few shot learning - Places205

places205 low-shot svm conv1

places205 low-shot svm conv2

places205 low-shot svm conv3

places205 low-shot svm conv4

places205 low-shot svm conv5
Image Classification

Image classification
Places-205, VOC’07

Object detection
VOC’07

3D Understanding
Surface Normals – NYUv2

Navigation
Gibson environment
Linear SVMs on VOC07

VOC2007 SVM classification. ResNet50

<table>
<thead>
<tr>
<th>Init</th>
<th>conv1</th>
<th>stage1</th>
<th>stage2</th>
<th>stage3</th>
<th>stage4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageNet supervised</td>
<td>24.49</td>
<td>47.75</td>
<td>60.54</td>
<td>80.36</td>
<td>87.95</td>
</tr>
<tr>
<td>ImageNet jigsaw</td>
<td>27.09</td>
<td>45.73</td>
<td>56.61</td>
<td>64.51</td>
<td>57.17</td>
</tr>
<tr>
<td>Imagenet14M jigsaw</td>
<td>23.46</td>
<td>46.72</td>
<td>58.52</td>
<td>71.76</td>
<td>64.92</td>
</tr>
<tr>
<td>YFCC100M jigsaw</td>
<td>18.98</td>
<td>46.71</td>
<td>57.77</td>
<td>71.21</td>
<td>64.34</td>
</tr>
</tbody>
</table>

- Deeper self supervised layers are less transferable
- Hypothesis – Problem is not “complex” enough.
SGD-based Linear Classifiers on Places205

Places205 linear classification. ResNet50

<table>
<thead>
<tr>
<th>Init</th>
<th>conv1</th>
<th>stage1</th>
<th>stage2</th>
<th>stage3</th>
<th>stage4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageNet supervised</td>
<td>14.84</td>
<td>32.59</td>
<td>42.06</td>
<td>50.83</td>
<td>52.49</td>
</tr>
<tr>
<td>ImageNet jigsaw</td>
<td>15.079</td>
<td>28.753</td>
<td>36.825</td>
<td>41.232</td>
<td>34.364</td>
</tr>
<tr>
<td>Imagenet14M jigsaw</td>
<td>13.973</td>
<td>29.462</td>
<td>36.656</td>
<td>41.721</td>
<td>36.254</td>
</tr>
<tr>
<td>YFCC100M jigsaw</td>
<td>10.252</td>
<td>29.672</td>
<td>39.565</td>
<td>44.866</td>
<td>38.219</td>
</tr>
</tbody>
</table>

- Gap between ImageNet and self-supervised methods is smaller.
- Places has scenes while ImageNet is object centric
- YFCC is a good mixture of both.
Evaluation: Main Lessons

- Evaluation on multiple tasks is essential
- Evaluation with fixed features or at least same hyper-parameters
- Evaluate sample efficiency of representations
What's missing from self-supervised methods?

- Complex problems, big data and deeper models
- Current self-supervised methods do not seem to learn high level representations
- Sample efficiency
Thanks!

Image classification
Few-shot learning
ImageNet, Places-205, VOC’07, COCO

Object detection
VOC’07

3D Understanding
Surface Normals – NYUv2

Navigation
Gibson environment